(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété Intellectuelle
Bureau international

(43) Date de la publication internationale
19 février 2004 (19.02.2004)

PCT

(10) Numéro de publication internationale
WO 2004/015571 A2

(51) Classification internationale des brevets[7] : G06F 9/46

(21) Numéro de la demande internationale :
PCT/FR2003/002466

(22) Date de dépôt international : 5 août 2003 (05.08.2003)

(25) Langue de dépôt : français

(26) Langue de publication : français

(30) Données relatives à la priorité :
02/10000 6 août 2002 (06.08.2002) FR

(71) Déposant *(pour tous les États désignés sauf US)* : CHECKFLOW [FR/FR]; 68, rue du Faubourg Saint-Honoré, F-75008 Paris (FR).

(72) Inventeurs; et
(75) Inventeurs/Déposants *(pour US seulement)* : BONNET, Vincent [FR/FR]; 21, rue Castelnau, F-94550 Chevilly-Larue (FR). PIASEK, Serge [FR/FR]; 25, rue du Temple, F-75004 Paris (FR). PROVOST, Lionel [FR/FR]; 93, rue de la Santé, F-75013 Paris (FR).

(74) Mandataires : BREESE, Pierre etc.; Breese-Majerowicz, 3, avenue de l'Opéra, F-75001 Paris (FR).

(81) États désignés *(national)* : AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) États désignés *(régional)* : brevet ARIPO (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), brevet eurasien (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), brevet européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO, SE, SI, SK, TR), brevet OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Publiée :
— *sans rapport de recherche internationale, sera republiée dès réception de ce rapport*

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

(54) Title: METHOD OF COMMUNICATING BETWEEN APPLICATIONS WHICH IS INTENDED TO SECURE ACCESS TO APPLICATION DATA

(54) Titre : PROCEDE DE COMMUNICATION ENTRE APPLICATIONS DESTINE A SECURISER L'ACCES AUX DONNEES D'UNE APPLICATION

(57) Abstract: The invention relates to a method of communicating between at least two applications, A and B, in an operating system, which is intended to prevent application A from accessing the information content of a window of application B. The inventive method comprises the following steps: creation of at least one variable by application B, receipt of a request from application A by application B, verification of the value of the aforementioned variable by application B in order to verify the validity of the request or authenticate the origin thereof and response to the request according to the value and/or origin. In this way, the invention is intended to solve the problem associated with securing personal data which the user provides to an application and which are normally transmitted to any other application requesting same. In particular, the invention can be used to secure data contained in a Web browser.

(57) Abrégé : La présente invention concerne un procédé de communication entre au moins deux applications A et B dans un système d'exploitation destiné à empêcher l'application A d'accéder au contenu des informations d'une fenêtre de l'application B comprenant les étapes suivantes: une étape de création d'au moins une variable par l'application B; une étape de réception d'une requête de l'application A par l'application B; une étape de vérification de la valeur de ladite variable par l'application B dans le but de vérifier la validité de ladite requête, ou d'authentifier sa provenance; une étape de réponse à ladite requête en fonction de ladite valeur et/ou ladite provenance. L'invention entend ainsi répondre au problème de la sécurisation des données personnelles fournies par l'utilisateur à une application et transmises ordinairement à toute autre application qui les demande. En particulier, l'invention sert à sécuriser les données contenues dans un navigateur web.

WO 2004/015571 A2

PROCÉDÉ DE COMMUNICATION ENTRE APPLICATIONS DESTINÉ À SÉCURISER L'ACCÈS AUX DONNÉES D'UNE APPLICATION

La présente invention se rapporte au domaine de la communication entre applications au sein d'un système d'exploitation. En effet, dans les systèmes d'exploitation d'ordinateur standard (« Windows » (marque déposée)…), les applications lancées échangent des messages par le biais du système pour obtenir des renseignements les unes sur les autres.

La présente invention entend ainsi répondre au problème de la confidentialité sur Internet en interdisant à certaines ou toutes les applications d'avoir accès aux données utilisateurs recueillies par un navigateur par exemple.

En effet, de plus en plus, surfer sur Internet sans être espionné est une illusion. De nombreux logiciels « gratuits » disponibles sur Internet se servent de l'accès que l'utilisateur leur accorde en les installant pour espionner les connexions de celui-ci et dresser un profil consommateur à revendre. Pire, certains logiciels ont pour but de ramener à leurs créateurs notamment les mots de passe, les identifiants, le numéro de carte de crédit ou tout autre information personnelle de l'utilisateur. La méthode utilisée par ces logiciels espions (« spywares » en anglais) est simple : la plupart des systèmes d'exploitation étant faits pour que les applications puissent dialoguer entre elles, ces logiciels espions demandent simplement au navigateur l'adresse du site, ou la valeur de certains champs d'une page web (en mode Secure Socket Layer, ou non...) remplis par l'utilisateur et le navigateur leur fournit cette information.

L'art antérieur connaît déjà par le brevet américain US6000032 un dispositif et une méthode pour

obtenir une valeur de sécurité qui permet à un module appelant d'accéder de manière sécurisée à un module appelé dans un ordinateur numérique. Ce dispositif permet d'accorder l'accès à un module logiciel seulement sous présentation d'une valeur prédéfinie. Cependant, le problème résolu par ce dispositif est la protection d'un système logiciel aux attaques hostiles tout en autorisant les interlocuteurs identifiés d'accéder aux données. La méthode met en œuvre des calculs relativement compliqués destinés à déterminer les droits du module appelant. Cette invention de l'art antérieur ne répond donc pas au même problème technique et la solution qu'elle propose est trop compliquée à mettre en place pour le problème que la présente invention entend résoudre.

D'autre part, une solution connue consiste à développer des alternatives à des applications fortement répandues de manière à profiter de l'ignorance de ces nouvelles applications par les logiciels espions. Cette solution comporte comme limite principale et fondamentale que lorsque l'alternative devient connue, les développeurs des logiciels espions l'intègrent dans la liste des applications avec lesquelles ceux-ci peuvent communiquer.

La présente invention entend remédier aux inconvénients de l'art antérieur en proposant un système utilisant les messages inter-applications standard du système d'exploitation pour effectuer un contrôle d'accès à ses données par une application.

Pour ce faire, la présente invention est du type décrit ci-dessus et elle est remarquable dans son acceptation la plus large, en ce qu'elle concerne un procédé de communication entre au moins deux applications A et B dans un système d'exploitation destiné à empêcher

l'application B d'accéder au contenu des informations d'une fenêtre de l'application A caractérisé en ce qu'il comprend les étapes suivantes :

- une étape de création d'au moins une variable par l'application A ;
- une étape de réception d'une requête de l'application B par l'application A ;
- une étape de vérification de la valeur de ladite variable par l'application A dans le but de vérifier la validité de ladite requête, ou d'authentifier sa provenance;
- une étape de réponse à ladite requête en fonction de ladite valeur et/ou ladite provenance.

Dans un cas particulier de l'invention, les deux applications A et B sont la même, c'est-à-dire que A est égal à B. Le procédé comprend alors une étape additionnelle consistant à modifier la valeur de la variable pour que ladite requête soit considérée valide.

Avantageusement, l'étape de vérification est réalisée par une fonction du système d'exploitation surchargée.

De préférence, le système d'exploitation est « Microsoft Windows » (marque déposée) mais il peut être tout autre système d'exploitation apte à utiliser/gérer des messages entre applications.

Selon un mode de réalisation de l'invention, ladite valeur vérifiée par l'application A est différente d'une valeur prédéfinie et l'étape de réponse consiste à ne pas satisfaire ladite requête.

Selon un autre mode de mise en œuvre, ladite valeur vérifiée par l'application A est égale à une valeur prédéfinie et l'étape de réponse consiste à satisfaire ladite requête.

On comprendra mieux la présente invention à l'aide de la description, faite ci-après à titre purement

explicatif, d'un mode de réalisation de l'invention, en référence aux figures annexées :

- La figure 1 illustre le processus standard de communication entre deux applications ;
- La figure 2 illustre le processus de communication entre deux applications selon l'invention.

Selon un mode de réalisation préféré de l'invention, celle-ci concerne le système d'exploitation « Windows » (marque déposée) dans ses versions les plus répandues. Dans ce système d'exploitation, une application A, qui peut être un logiciel de messagerie instantanée doté d'un logiciel espion, cherche à récupérer la valeur du champ URL d'une fenêtre d'une application B qui peut être par exemple un navigateur Internet.

Dans un système d'exploitation standard, les applications communiquent selon le procédé décrit ci-dessous et illustré figure 1.

Lors de l'étape (1), une application A adresse un message à une application B afin d'obtenir des informations sur des éléments de l'application B.

L'étape (2) consiste pour l'application B ou une de ses fonctions internes à traiter le message.

L'étape (3) est la réponse de l'application B à l'application A par la fourniture des informations demandées.

Dans un système comprenant une application B dotée du procédé selon l'invention, les communications entre une autre application A et ladite application B sont illustrés figure 2.

Lors de l'étape (4), une application A adresse un message à une application B afin d'obtenir des informations sur des éléments de l'application B.

L'étape (5) consiste pour l'application B ou une de ses fonctions internes à traiter le message en fonction de la valeur d'une variable interne à l'application B au moment du traitement du message.

Si la valeur autorise la réponse au message, l'application B répond à l'application A de la même manière que dans le procédé standard (étape 6).

Sinon, l'application B ne répond pas à l'application A mais signifie au système d'exploitation que le message a été traité (étape 7).

Un mode de réalisation particulier est décrit ci-dessous dans le système d'exploitation « Microsoft Windows » (marque déposée).

Une application A désirant obtenir des informations d'une application B génère une commande « send_message » avec pour paramètres le type du message et l'identité du destinataire. Si l'application cible B est un navigateur et que l'application cherche à obtenir le contenu du champ URL de l'application B, le type du message sera CB_GETLBTEXT et l'identité du destinataire sera l'identifiant de la fenêtre de l'application cible B. Cette commande induit la création d'une variable dans les registres de l'application A destinée à recueillir la réponse de l'application visée ainsi que l'envoi d'un message au système d'exploitation contenant l'adresse de la variable de registre et l'identité du destinataire.

Le système d'exploitation reçoit le message de l'application A et envoie l'adresse de la variable de registre à la fenêtre de l'application B visée.

Lors de la création de la fenêtre cible, une fonction de traitement a été attribuée à cette fenêtre destinée en particulier à traiter les messages destinés à cette fenêtre. Cette fonction est appelée

« DefWindowProc() » dans « Microsoft Windows » (marque déposée). Les messages sont donc disposés dans une pile.

Pour chaque message de la pile, la fonction de traitement lit le message et y répond en remplissant des champs vierges du registre de l'application A créés lors de l'émission du message par celle-ci. Dans certaines versions de « Microsoft Windows » (marque déposée), ces champs sont appelés « wparam » et « lparam », et contiennent d'une part la longueur de la réponse et d'autre part la réponse. Dans « Microsoft Windows » (marque déposée), si la requête consistait à obtenir la valeur du champ URL du navigateur, la fonction de traitement de la fenêtre lirait la valeur contenue pour la variable « ComboBoxEx » (correspondant audit champ URL) pour donner la réponse.

L'application A lit la réponse enregistrée dans ses registres puis supprime la variable créée pour ce message.

Le procédé selon l'invention consiste à réaliser les opérations suivantes.

Avant toute communication et au lancement de l'application cible B à :

- créer et initialiser (à 0) une variable de provenance dans les registres de l'application B destinée à déterminer ultérieurement la provenance des messages entrants.
- Lors de la création d'une fenêtre, à créer une nouvelle fonction de traitement similaire à celle créée par défaut et à surcharger cette nouvelle fonction de traitement afin de déterminer la provenance des messages entrants.

Lorsque l'application B envoie un message à destination d'elle-même, avant l'envoi du message, la variable de provenance est mise à 1.

Dans tous les cas, un message à destination de la fenêtre de l'application B est transmis d'abord à la nouvelle fonction de traitement.

A la réception d'un message, la nouvelle fonction de traitement scrute tout d'abord le registre contenant la variable de provenance et lit la valeur de celle-ci. Si cette valeur est égale à 1, le message est transmis à la fonction de traitement standard de la fenêtre qui termine le traitement selon le procédé décrit ci-dessus. Si la variable de provenance est à 0, le message n'est pas traité et l'application B envoie un message au système d'exploitation pour lui signifier le traitement du message est terminé.

L'invention est décrite dans ce qui précède à titre d'exemple. Il est entendu que l'homme du métier est à même de réaliser différentes variantes de l'invention sans pour autant sortir du cadre du brevet.

REVENDICATIONS

1. Procédé de communication entre au moins deux applications A et B dans un système d'exploitation destiné à empêcher l'application A d'accéder au contenu des informations d'une fenêtre de l'application B, caractérisé en ce qu'il comprend les étapes suivantes :

- une étape de création d'au moins une variable par l'application B ;
- une étape de réception d'une requête de l'application A par l'application B ;
- une étape de vérification de la valeur de ladite variable par l'application B dans le but de vérifier la validité de ladite requête, ou d'authentifier sa provenance ;
- une étape de réponse à ladite requête en fonction de ladite valeur et/ou ladite provenance.

2. Procédé selon la revendication 1, caractérisé en ce que les deux applications A et B sont la même, c'est-à-dire que A est égal à B.

3. Procédé selon la revendication 2, caractérisé en ce que le procédé comprend une étape additionnelle consistant à modifier la valeur de la variable pour que ladite requête soit considérée valide.

4. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que l'étape de vérification est réalisée par une fonction du système d'exploitation surchargée.

5. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que le système

d'exploitation est apte à utiliser et gérer des messages entre applications.

6. Procédé selon l'une quelconque des revendications précédentes, caractérisé en ce que ladite valeur vérifiée par l'application B est différente d'une valeur prédéfinie et que l'étape de réponse consiste à ne pas satisfaire ladite requête.

7. Procédé selon l'une quelconque des revendications 1 à 5, caractérisé en ce que ladite valeur vérifiée par l'application B est égale à une valeur prédéfinie et que l'étape de réponse consiste à satisfaire ladite requête.

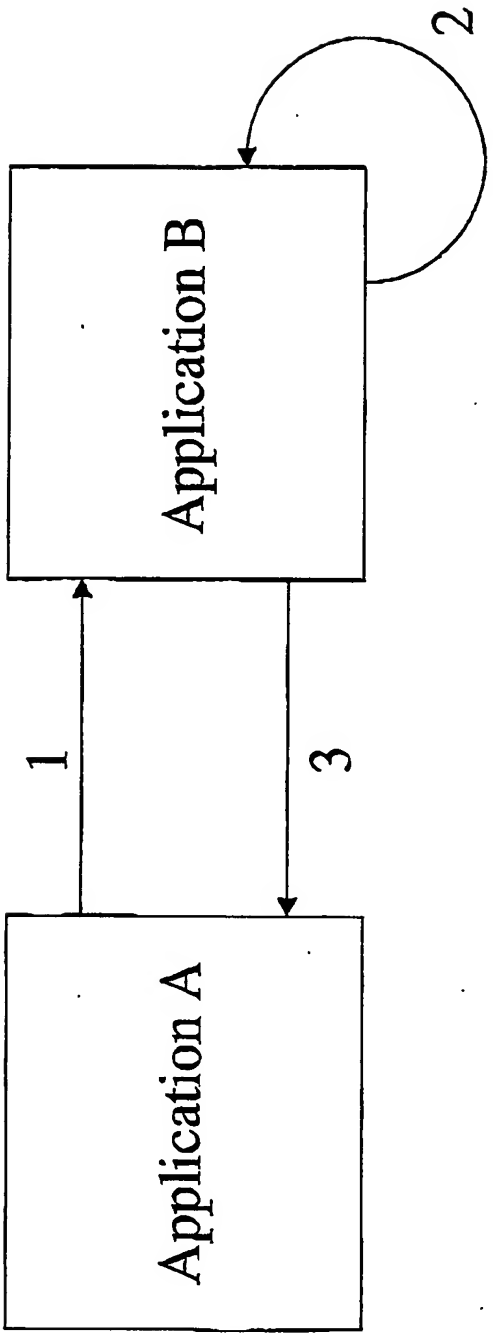

Figure 1

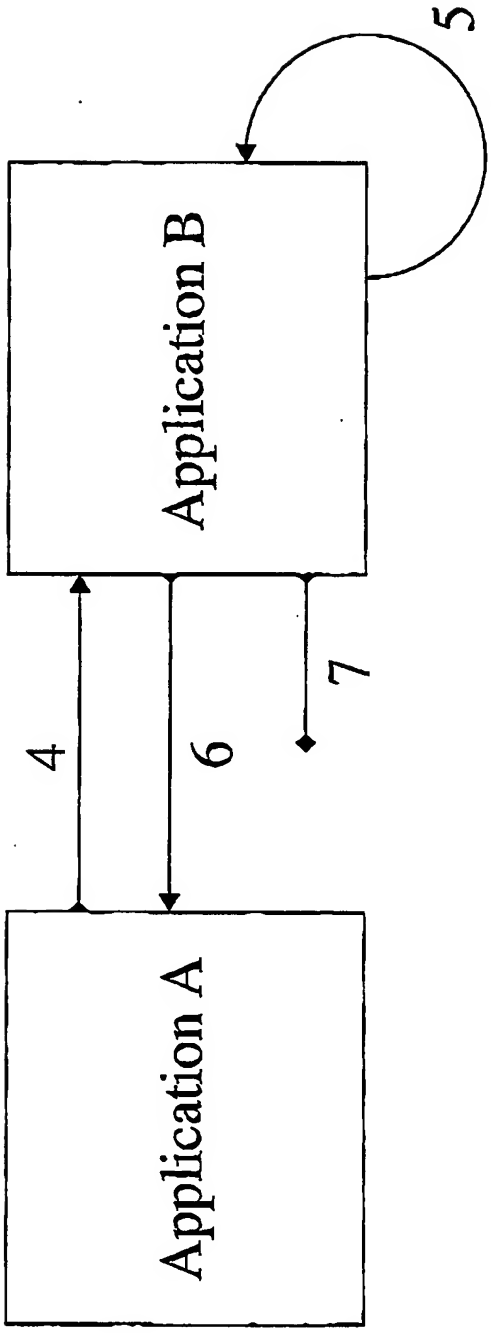

Figure 2